



(11) **EP 4 354 242 A1**

(12) **EUROPEAN PATENT APPLICATION**
published in accordance with Art. 153(4) EPC

(43) Date of publication:
17.04.2024 Bulletin 2024/16

(51) International Patent Classification (IPC):
G05B 23/02 (2006.01)

(21) Application number: **22885424.6**

(52) Cooperative Patent Classification (CPC):
G05B 23/02; Y02P 90/02

(22) Date of filing: **05.09.2022**

(86) International application number:
PCT/CN2022/117083

(87) International publication number:
WO 2023/071534 (04.05.2023 Gazette 2023/18)

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME
Designated Validation States:
KH MA MD TN

• **GD Midea Heating & Ventilating Equipment Co., Ltd.**
Foshan, Guangdong 528311 (CN)

(72) Inventors:
• **HUANG, Wentian**
Shanghai 200080 (CN)
• **YAN, Jie**
Shanghai 200080 (CN)

(30) Priority: **01.11.2021 CN 202111283676**

(74) Representative: **RGTH**
Patentanwälte PartGmbB
Neuer Wall 10
20354 Hamburg (DE)

(71) Applicants:
• **Shanghai Kong Intelligent Building Co., Ltd**
Shanghai 200080 (CN)

(54) **REMOTE DEBUGGING METHOD AND APPARATUS**

(57) Provided are a remote debugging method and apparatus, an electronic device, a computer-readable storage medium, a computer program product, and a computer program. The method comprises: receiving first operation data of a controller (S101); according to a debugging instruction, acquiring the first operation data

of the controller to be debugged, wherein the debugging instruction comprises identification information of the controller to be debugged (S102); and generating debugging data according to the first operation data, and sending the debugging data to the controller to be debugged (S 103).

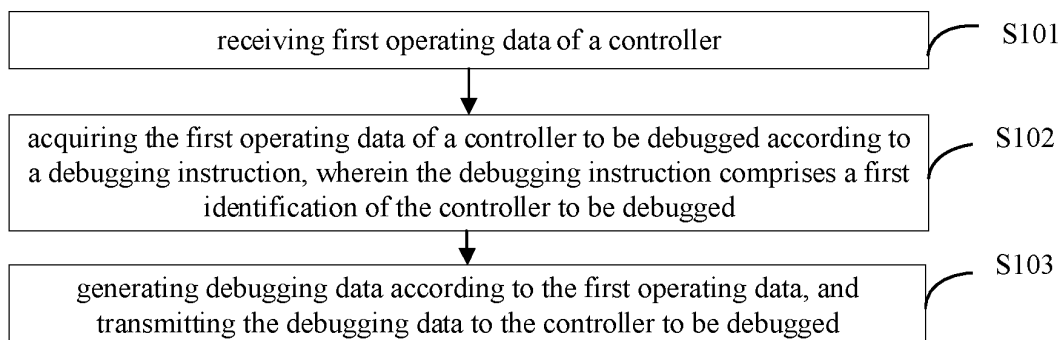


Figure 1

EP 4 354 242 A1

Description

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The application is based on and claims priority to Chinese Patent Application No. 202111283676.2, filed on November 1, 2021, the entire content of which is incorporated herein by reference.

FIELD

[0002] The present disclosure relates to the technical field of intelligent control, and specifically to a remote debugging method and device therefor, an electronic device, computer-readable storage medium, computer program product and computer program.

BACKGROUND

[0003] A controller belongs to the building automatic control product, where the building equipment may be subjected to some adjustments and debugging according to the owner's demand. In the related art, when debugging the controller, technicians need to debug on site, which however are unable to arrive at the site for debugging at once when the customer has demand. It is not in time and of high cost.

SUMMARY

[0004] The present disclosure aims to solve at least one of the technical problems in the related art to a certain degree.

[0005] According to a first aspect of embodiments of the present disclosure, there is provided a remote debugging method, for a cloud server side, comprising: receiving first operating data of a controller; acquiring the first operating data of a controller to be debugged according to a debugging instruction, wherein the debugging instruction comprises an identification of the controller to be debugged; and generating debugging data according to the first operating data, and transmitting the debugging data to the controller to be debugged.

[0006] According to a second aspect of embodiments of the present disclosure, there is provided another remote debugging method, for a client side, comprising: transmitting first operating data of a controller; and receiving debugging data corresponding to a controller to be debugged, and debugging the controller to be debugged according to the debugging data.

[0007] According to a third aspect of embodiments of the present disclosure, there is provided a remote debugging device, for a cloud server side, comprising: a first receiving module, configured to receive first operating data of a controller; a data acquiring module, configured to acquire the first operating data of a controller to be debugged according to a debugging instruction, wherein the debugging instruction comprises an identifi-

cation of the controller to be debugged; and a data transmitting module, configured to generate debugging data according to the first operating data, and transmit the debugging data to the controller to be debugged.

[0008] According to a fourth aspect of embodiments of the present disclosure, there is provided another remote debugging device, for a client side, comprising: a transmitting module, configured to transmit first operating data of a controller; and a receiving module, configured to receive debugging data corresponding to a controller to be debugged, and debug the controller to be debugged according to the debugging data.

[0009] According to a fifth aspect of embodiments of the present disclosure, there is provided an electronic device, comprising: at least one processor; and a memory, communicated to said at least one processor, wherein the memory has stored therein executable instructions that, when executed by said at least one processor, achieve a remote debugging method according to the first or second aspect of embodiments of the present disclosure.

[0010] According to a sixth aspect of embodiments of the present disclosure, there is provided a non-transitory computer-readable storage medium having stored therein computer executable instructions that, when executed by a computer, achieves a remote debugging method according to the first or second aspect of embodiments of the present disclosure.

[0011] According to a seventh aspect of embodiments of the present disclosure, there is provided a computer program product, comprising a computer program that, when executed by a processor, achieves a remote debugging method according to the first or second aspect of embodiments of the present disclosure.

[0012] According to an eighth aspect of embodiments of the present disclosure, there is provided a computer program, comprising a computer program code that, when executed by a computer, implements a remote debugging method according to the first or second aspect of embodiments of the present disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013]

Figure 1 is a block diagram showing a remote debugging method according to at least one of examples of the present disclosure.

Figure 2 is a schematic diagram showing a remote debugging according to at least one of examples of the present disclosure.

Figure 3 is a block diagram showing a remote debugging method according to at least one of examples of the present disclosure.

Figure 4 is a block diagram showing transmission of debugging data to a controller to be debugged, according to at least one of examples of the present disclosure.

Figure 5 is a schematic diagram showing acquirement for debugging data, according to at least one of examples of the present disclosure.

Figure 6 is a block diagram showing reception to second operating data of a controller to be debugged, according to at least one of examples of the present disclosure.

Figure 7 is a schematic diagram showing generation of a target data architecture, according to at least one of examples of the present disclosure.

Figure 8 is a block diagram showing a remote debugging method according to at least one of examples of the present disclosure.

Figure 9 is a block diagram showing another remote debugging method according to at least one of examples of the present disclosure.

Figure 10 is a block diagram showing transmission of second operating data, according to at least one of examples of the present disclosure.

Figure 11 is a schematic diagram showing acquirement for second operating data, according to at least one of examples of the present disclosure.

Figure 12 is a schematic diagram showing a remote debugging device according to at least one of examples of the present disclosure.

Figure 13 is a schematic diagram showing a remote debugging device according to at least one of examples of the present disclosure.

Figure 14 is a schematic diagram showing an electronic device according to at least one of examples of the present disclosure.

DETAILED DESCRIPTION

[0014] Reference will be made in detail to embodiments of the present disclosure. The same or similar elements and the elements having same or similar functions are denoted by like reference numerals throughout the descriptions. The embodiments described herein with reference to drawings are illustrative, and used to generally understand the present disclosure. The embodiments shall not be construed to limit the present disclosure.

[0015] Figure 1 is a block diagram showing a remote debugging method provided in at least one of examples of the present disclosure. As shown in Figure 1, the remote debugging method comprises the following steps 5101-5103.

[0016] At the S101, first operating data of a controller is received.

[0017] There are generally a lot of environmental monitoring equipment and electromechanical equipment in the interior of buildings. The equipment belonging to a same kind builds an integrated subsystem, through a controller. In a building, there are generally many subsystems, such as security subsystem, air conditioning subsystem, cold and heat source subsystem, water supply and drainage subsystem, power transformation and

distribution subsystem, lighting subsystem, elevator subsystem and so on.

[0018] To achieve a remote debugging to the controller, the controller may transmit its first operating data in real time to a remote device capable of performing debugging. In some examples, the remote device capable of performing debugging may be a server, a cloud, a control terminal, a debugging terminal, etc., that is, the server, the cloud, the control terminal, the debugging terminal, etc. receive the first operating data of the controller.

[0019] At the S102, the first operating data of a controller to be debugged is acquired according to a debugging instruction, where the debugging instruction comprises an identification of the controller to be debugged.

[0020] In some examples, the controller may transmit the first operating data, by itself, to the remote device capable of performing debugging. In some examples, the controller may be a direct digital control (DDC) controller, a programmable logic controller (PLC) and so on.

[0021] In other examples, the controller may transmit the first operating data to a client side, and the client side unifies and manages the controller, and transmits the first operating data to the remote device capable of performing debugging.

[0022] The remote device capable of performing debugging such as the server, the cloud, the control terminal, the debugging terminal, etc., after receiving the debugging instruction, acquires the first operating data according to the debugging instruction and the identification of the controller to be debugged comprised in the debugging instruction. The debugging instruction may be generated according to external instructions of debugging personnel or customers, or be automatically generated according to internal debugging strategies.

[0023] At the S103, debugging data is generated according to the first operating data, and the debugging data is transmitted to the controller to be debugged.

[0024] The debugging personnel start to debug according to the received first operating data. In some examples, debugging may be interface dragging, compilation deployment, etc. Based on debugging by debugging personnel, the device performing debugging generates the debugging data and transmits the debugging data to the controller to be debugged.

[0025] The present disclosure provides in at least one of examples the remote debugging method, comprising: receiving the first operating data of a controller; acquiring the first operating data of a controller to be debugged according to the debugging instruction, where the debugging instruction comprises the identification of the controller to be debugged; and generating the debugging data according to the first operating data, and transmitting the debugging data to the controller to be debugged. According to examples of the present disclosure, the device performing debugging may acquire real-time information of several controllers, such that the debugging personnel can remotely acquire the information of the controllers through the device performing debugging to

remotely debug the controller, which is more convenient and faster with reduced costs.

[0026] Figure 2 is a schematic diagram showing a remote debugging system according to at least one of examples of the present disclosure. The remote debugging system comprises: a plurality of cloud servers, a plurality of client sides and a plurality of controllers, where one cloud server may correspond to one or more client side(s), each of which manages one or more controller(s).

[0027] In some examples, the controller under the management of the client side may be a direct digital control (DDC) controller (known as DDC controller). When the controller under the management of the client side is a DDC controller, a DDC system may comprise central control equipment, field DDC controller, communication network, and corresponding sensors, actuators, regulating valves and other components. There are generally a lot of environmental monitoring equipment and electromechanical equipment in the interior of buildings. The equipment belonging to a same kind builds an integrated subsystem, through a DDC controller. In a building, there are generally many subsystems, such as security subsystem, air conditioning subsystem, cold and heat source subsystem, water supply and drainage subsystem, power transformation and distribution subsystem, lighting subsystem, elevator subsystem and so on.

[0028] The remote debugging method provided in examples of the present disclosure is illustrated below, by taking the remote debugging system as shown in Figure 2 as an example.

[0029] Figure 3 is a block diagram showing a remote debugging method according to at least one of examples of the present disclosure. As shown in Figure 3, the remote debugging method comprises the following steps 5301-5303.

[0030] At the S301, the first operating data of each controller is received from each client side managing the controller respectively.

[0031] As shown in Figure 2, the controller may manage the monitoring subsystem which is under the control of the controller, and transmit the first operating data representing an operating state of the monitoring subsystem to the client side, which transmits the first operating data sent by the controller to the cloud server, thereby realizing intensive management of the client side. The first operating data comprises resource usage state data and/or configuration information of the controller to be debugged. For example, the configuration information may comprise html, css, js, svg, icon, etc., and the resource usage state data may comprise real-time data of hardware. In some examples, the cloud server may implement a two-level storage mechanism. On the one hand, the binding relationship between the cloud server and the client side and the controller managed by the client side would be persisted locally, thereby forming a mapping relationship that can be read on demand and can prevent from loss resulting from power failure. On the

other hand, the cloud server would store this binding relationship in memory to achieve high-speed reading and high-speed matching.

[0032] In some examples, when the cloud server receives the first operating data transmitted by the client side, heartbeat data may be sent to the client side at set intervals to keep the pipeline alive.

[0033] At the S302, the first operating data of the controller to be debugged is acquired according to the debugging instruction, where the debugging instruction comprises an identification of the controller to be debugged.

[0034] When the controller requires debugging, the debugging personnel may communicate with the cloud server through a terminal device, for example, through the browser of the terminal device, and transmit the debugging instruction for debugging the controller to be debugged to the cloud server, and the cloud server receives the debugging instruction sent by the debugging personnel through the browser of the terminal device. The debugging instruction comprises identification of the controller to be debugged, such as an IP address of the controller to be debugged. In some examples, the terminal device may be a mobile phone, tablet, computer, etc.

[0035] The cloud server, according to the received debugging instruction and the identification of the controller to be debugged therein, extracts the first operating data corresponding to the controller to be debugged. The first operating data comprises resource usage state data and/or configuration information of the controller to be debugged. For example, the configuration information may comprise html, css, js, svg, icon, etc., and the resource usage state data may comprise real-time data of hardware.

[0036] At the S303, debugging data is generated according to the first operating data, and the debugging data is transmitted to the controller to be debugged.

[0037] The debugging personnel start to debug according to the first operating data as received. In some examples, debugging may be interface dragging, compilation deployment, etc. Based on debugging of debugging personnel, the cloud server generates the debugging data and transmits the debugging data to a target client side to which the controller to be debugged belongs, and the target client side transmits the debugging data to the controller to be debugged.

[0038] The present disclosure provides in examples the remote debugging method, comprising: receiving the first operating data of each controller from each client side managing the controller respectively; acquiring the first operating data of a controller to be debugged according to the debugging instruction, where the debugging instruction comprises the identification of the controller to be debugged; and generating the debugging data according to the first operating data, and transmitting the debugging data to the controller to be debugged. According to examples of the present disclosure, the cloud server may acquire real-time information of the

controller managed by several client sides, such that the debugging personnel can remotely acquire the information of the controllers from the cloud server through the terminal device, to remotely debug the controller, which is more convenient and faster with reduced costs.

[0039] In some examples, when there are multiple cloud servers, all cloud servers may be grouped into a cloud server cluster, and a balancing server is set in the cloud server cluster. Debugging personnel send the debugging instruction to the balancing server in the cloud server cluster, which determines a cloud server for processing the debugging instruction according to load status of each cloud server in the cluster, and transmits the debugging instruction to the determined cloud server, thereby realizing a load balance in the cloud server cluster. For example, if there are three cloud servers and one balancing server in the cloud server cluster and two of them are working, the balancing server can distribute the debugging instruction to the third cloud server.

[0040] In some examples, after receiving the first operating data transmitted by the server, it is judged whether the first operating data carries the project identification and the client's identification; and if the first operating data does not carry the project identification or the client's identification, a filter may be set on the cloud server, where the filter searches for the project identification or the client's identification based on the anti-theft chain technology in Hyper Text Transfer Protocol (HTTP), and adds the project identification and/or the identification into the first operating data. For example, the project information corresponding to the controller to be debugged may be "*** Park", and the identification of the client side to which the controller to be debugged belongs may be an IP address of the client side.

[0041] Figure 4 is a block diagram showing a remote debugging method according to at least one of examples of the present disclosure, which is executed by a cloud server side. As shown in Figure 4, the transmission of debugging data to a target client side to which a controller to be debugged belongs comprises the following steps 5401-5402.

[0042] At the S401, a debugging link of the controller to be debugged is determined according to the identification of the controller to be debugged.

[0043] The project identification of the controller to be debugged and the identification of the client side to which the controller to be debugged belongs are determined according to the identification of the controller to be debugged; and the debugging link is generated based on the identification and the project identification of the controller to be debugged, and the identification of the target client side to which the controller to be debugged belongs. For example, the project information corresponding to the controller to be debugged may be "*** Park", the identification of the client side to which the controller to be debugged belongs may be an IP address of the client side, and the identification of the controller to be debugged may be an IP address of the controller to be

debugged.

[0044] At the S402, the debugging data is transmitted to the controller to be debugged through the debugging link.

5 **[0045]** Figure 5 is a schematic diagram showing acquirement for debugging data. As shown in Figure 5, an interceptor is set on the cloud server for judging a debugging type of the debugging instruction, for example, the debugging type may be a data type or a data-free type, where the data type may be divided into a normal data type and a byte stream data type. Based on the debugging type, a method base required to be called for the debugging type is acquired, and a data architecture, for debugging correspondingly, as the debugging data, is generated. After acquiring the debugging data, the debugging data is transmitted to the target client side through the debugging link, and further transmitted to the controller to be debugged, thereby realizing debugging for the controller to be debugged.

20 **[0046]** In some examples, the method base may comprise a P method base and a G method base. The method base automatically identifies whether the debugging type is the data type or data-free type and a request mode. If it is a get request, it will enter the G method base, which has methods calling html, svg, png, etc.; if it is a Post request, it will enter the P method base, which contains user-configured methods, time methods, node methods, language methods, etc.

25 **[0047]** According to at least one examples of the present disclosure, by the method the debugging link of each controller is determined, such that the cloud server can accurately and quickly find the target client side, so as to complete the debugging of the controller.

30 **[0048]** Figure 6 is a block diagram showing a remote debugging method according to at least one of examples of the present disclosure, which is executed by a cloud server side. As shown in Figure 6, subsequent to transmitting the debugging data to the target client side to which the controller to be debugged belongs, the method further comprises the following steps 5601-5602.

35 **[0049]** At the S601, second operating data generated, by the controller to be debugged, during and/or after a debugging process, is received from the target client side.

40 **[0050]** The second operating data generated, by the controller to be debugged, during and/or after the debugging process, is received from the target client side. The second operating data comprises resource usage state data and/or configuration information of the controller to be debugged. For example, the configuration information may comprise html, css, js, svg, icon, etc., and the resource usage state data may comprise real-time data of hardware.

45 **[0051]** After receiving the second operating data transmitted by the server, it is judged whether the second operating data carries the project identification and the client's identification; and if the second operating data does not carry the project identification or the client's identi-

cation, the project identification or the client's identification may be searched based on the anti-theft chain technology in HTTP, and added into the first operating data.

[0052] Figure 7 is a schematic diagram showing generation of a target data architecture. As shown in Figure 7, the received second operating data is collated for removing unnecessary information and retain key information, thereby reducing the amount of data transmitted. The second operating data with unnecessary information removed is adjusted and cleaned structurally on a hierarchical structure of authorization code, information abstract, event type and event, and finally converted into a relatively fixed data structure through filters, adapters and decorators, to generate the target data architecture.

[0053] At the S602, a debugging log of the controller to be debugged is generated according to the second operating data.

[0054] The debugging log of the debugged controller is generated according to the second operating data which is transmitted by the client side and generated during and/or after the debugging process. The debugging log may record debugging requesting data, debugging data and debugging result data.

[0055] According to at least one of examples of the present disclosure, the cloud server receives the second operating data generated, by the controller to be debugged, during and/or after the debugging process, and generates the debugging log, which is beneficial for debugging personnel to monitor information of the debugged controller in real time, and the log recorded in the cloud server facilitates subsequent reference when required.

[0056] Continuing to take the debugging structure of Figure 2 as an example, Figure 8 is a schematic diagram of a remote debugging method according to at least one of examples of the present disclosure, which is executed by a cloud server side. As shown in Figure 8, the remote debugging method comprises the following steps 5801 to 5810.

[0057] At the 5801, the first operating data of each controller is received from each client side managing the controller respectively.

[0058] At the 5802, the project identification of the controller to be debugged and the identification of the client side to which the controller to be debugged belongs are determined according to the identification of the controller to be debugged.

[0059] At the S803, the debugging link is generated based on the identification and the project identification of the controller to be debugged, and the identification of the target client side

[0060] The steps 5801-5803 may apply to implementation models according to examples of the present disclosure, which will be not repeated herein.

[0061] At the 5804, the debugging instruction is received from the terminal device, where the debugging instruction comprises the identification of the controller to be debugged.

[0062] At the 5805, the first operating data of the controller to be debugged is acquired according to the debugging instruction.

[0063] At the 5806, the debugging data is generated according to the first operating data.

[0064] At the S807, the debugging link of the controller to be debugged is determined according to the identification of the controller to be debugged.

[0065] At the 5808, the debugging data is transmitted to the controller to be debugged through the debugging link.

[0066] The steps 5804-5808 may apply to implementation models according to examples of the present disclosure, which will be not repeated herein.

[0067] At the S809, the second operating data generated, by the controller to be debugged, during and/or after a debugging process, is received from the target client side.

[0068] At the S810, the debugging log of the controller to be debugged is generated according to the second operating data.

[0069] The steps 5809-5810 may apply to implementation models according to examples of the present disclosure, which will be not repeated herein.

[0070] The present disclosure further provides in examples the remote debugging method, comprising: receiving the first operating data of each controller from each client side managing the controller respectively; acquiring the first operating data of a controller to be debugged according to the debugging instruction, where the debugging instruction comprises the identification of the controller to be debugged; and generating the debugging data according to the first operating data, and transmitting the debugging data to the target client side to which the controller to be debugged belongs. According to examples of the present disclosure, the cloud server may acquire real-time information of the controller managed by several client sides, such that the debugging personnel can remotely acquire the information of the controllers from the cloud server through the terminal device, to remotely debug the controller, which is more convenient and faster with reduced costs.

[0071] An executive subject of the remote debugging method provided by the present disclosure may directly interact with the controller to be debugged with data and/or information. If the executive subject is a client side, data and/or information interaction may be performed with the controller to be debugged. If the executive subject is a cloud server, data and/or information interaction may be performed with the controller to be debugged through the client side.

[0072] In some examples, subsequent to transmitting the debugging data to the controller to be debugged, the method further comprises: receiving the second operating data generated, by the controller to be debugged, during and/or after the debugging process; and generating the debugging log of the controller to be debugged according to the second operating data.

[0073] In some examples, transmitting the debugging data to the controller to be debugged comprises: determining the debugging link of the controller to be debugged according to the identification of the controller to be debugged; and transmitting the debugging data to the controller to be debugged through the debugging link.

[0074] In some examples, subsequent to acquiring the first operating data of a controller to be debugged, the remote debugging method further comprises: determining the project identification of the controller to be debugged and the identification of a client side to which the controller to be debugged belongs, according to the identification of the controller to be debugged; and generating the debugging link based on the identification and the project identification of the controller to be debugged, and the identification of a target client side to which the controller to be debugged belongs.

[0075] In some examples, subsequent to acquiring the first operating data or the second operating data, the remote debugging method further comprises: judging whether the first operating data or the second operating data carries the project identification and the identification of the client side; and adding the project identification and/or the identification into the first operating data or the second operating data, in response to the determination that the project identification and/or the identification of the client side are/is not carried.

[0076] In some examples, prior to transmitting the debugging data to the controller to be debugged, the remote debugging method further comprises: acquiring the debugging type of the debugging instruction, acquiring the method base required to be called for the debugging type according to the debugging type, and generating the data architecture, for debugging correspondingly, as the debugging data.

[0077] In some examples, subsequent to acquiring the first operating data or the second operating data, the remote debugging method further comprises: collating the first operating data or the second operating data to obtain key information; and adjusting the key information structurally to generate the target data architecture.

[0078] Continuing to take the debugging structure of Figure 2 as an example, Figure 9 is a schematic diagram of a remote debugging method according to at least one of examples of the present disclosure, which is executed by a target client side. As shown in Figure 9, the remote debugging method comprises the following steps S901-S902.

[0079] At the S901, first operating data of each controller managed by the target client side is transmitted to a target cloud server.

[0080] The client side transmits the first operating data of each controller under its management in real time to the target cloud server. The first operating data comprises resource usage state data and/or configuration information of the controller to be debugged. For example, the configuration information may comprise html, css, js, svg, icon, etc., and the resource usage state data may

comprise real-time data of hardware.

[0081] At the S902, debugging data corresponding to a controller to be debugged is received from the target cloud server, and the controller to be debugged is debugged according to the debugging data.

[0082] When debugging personnel debugs the controller to be debugged, the client side receives the debugging data corresponding to the controller to be debugged from the target cloud server, and debugs the controller to be debugged according to the debugging data.

[0083] The present disclosure further provides in examples another remote debugging method, comprising: transmitting the first operating data of each controller managed by the target client side to the target cloud server; and receiving the debugging data corresponding to the controller to be debugged from the target cloud server, and debugging the controller to be debugged according to the debugging data. According to examples of the present disclosure, the cloud server may acquire real-time information of the controller managed by several client sides, such that the debugging personnel can remotely acquire the information of the controllers from the cloud server through the terminal device, to remotely debug the controller, which is more convenient and faster with reduced costs.

[0084] Continuing to take the debugging structure of Figure 2 as an example, Figure 10 is a schematic diagram of a remote debugging method according to at least one of examples of the present disclosure, which is executed by a target client side. As shown in Figure 10, subsequent to receiving the debugging data corresponding to the controller to be debugged from the target cloud server, the method further comprises the following steps 51001-51004.

[0085] At the S1001, the debugging data is parsed to generate discrete debugging data.

[0086] Figure 11 is a schematic diagram showing acquirement for second operating data. As shown in Figure 11, the debugging data, received by the client side from the cloud server, is parsed with a parser and a decipherer, so as to generate several discrete debugging data, which may divided into a byte stream/character stream and an identification set.

[0087] At the S1002, the discrete debugging data is processed and regrouped to generate the regrouped debugging data.

[0088] The several discrete debugging data is processed, and then regrouped by a decorator, so as to generate the regrouped debugging data.

[0089] At the S1003, a debugging type of the regrouped debugging data is acquired, an event library required to be called for the debugging type is acquired according to the debugging type, and second operating data of the controller to be debugged is generated.

[0090] The debugging type the regrouped debugging data corresponds to is acquired. For example, the regrouped debugging data may correspond to html, css, js, svg, icon, etc., or real-time data of hardware. As shown

in Figure 11, according to the debugging type, the event library required to be called for the debugging type is acquired, and initial operating data of the controller to be debugged is generated through a verifier, a filter, and an encoder. The initial operating data is split, decorated, encrypted and encoded to generate the second operating data finally. The second operating data comprises resource usage state data and/or configuration information of the controller to be debugged. The cloud server judges the debugging type, and according to different debugging types goes to different event libraries, which contains corresponding sub-events and an addressing mapping table preset in advance, and the identification of a target controller can be quickly and efficiently found through the addressing mapping table. The addressing mapping table is made according to the network topology structure prepared in advance. After the cloud server finds a target address, the target address is returned back to an initial result set which has tracing markers of relevant events. With the tracing marker, an original debugging instruction could be found in an addresser, forming a closed loop. In some examples, the addressing mapping table may be stored in the addresser, which implements a two-level mechanism, one persistent to disk and the other resident in memory, to achieve high speed reading and matching.

[0091] At the S1004, the second operating data is transmitted.

[0092] Subsequent to acquiring the second operating data, the second operating data generated, by the controller to be debugged, during and/or after the debugging process is transmitted to the target cloud server through debugging link.

[0093] According to at least one of examples of the present disclosure, the client side transmits the second operating data generated, by the controller to be debugged, during and/or after the debugging process to the target cloud server, which is beneficial for debugging personnel to monitor information of the debugged controller in real time.

[0094] In some examples, subsequent to receiving the debugging data corresponding to a controller to be debugged, and debugging the controller to be debugged according to the debugging data, the remote debugging method further comprises: transmitting the second operating data generated, by the controller to be debugged, during and/or after the debugging process.

[0095] In some examples, the operating data of the controller comprises resource usage state data and/or configuration information of the controller to be debugged.

[0096] In some examples, subsequent to receiving debugging data corresponding to a controller to be debugged, the remote debugging method further comprises: parsing the debugging data to generate discrete debugging data; processing and regrouping the discrete debugging data to generate regrouped debugging data; acquiring a debugging type of the regrouped debugging data, acquiring an event library required to be called for

the debugging type according to the debugging type, and generating second operating data of the controller to be debugged; and transmitting the second operating data.

[0097] Figure 12 is a schematic diagram showing a remote debugging device according to at least one of examples of the present disclosure. As shown in Figure 12, the remote debugging device 1200 is installed at a cloud server, comprising: a first receiving module 1201, a data acquiring module 1202 and a data transmitting module 1203.

[0098] The first receiving module 1201 is configured to receive first operating data of a controller.

[0099] The data acquiring module 1202 is configured to acquire the first operating data of a controller to be debugged according to a debugging instruction, where the debugging instruction comprises an identification of the controller to be debugged.

[0100] The data transmitting module 1203 is configured to generate debugging data according to the first operating data, and transmit the debugging data to the controller to be debugged.

[0101] In some examples, the data transmitting module 1203 is further configured to receive second operating data generated, by the controller to be debugged, during and/or after a debugging process; and generate a debugging log of the controller to be debugged according to the second operating data.

[0102] In some examples, the operating data of the controller comprises resource usage state data and/or configuration information of the controller to be debugged.

[0103] In some examples, the data transmitting module 1203 is further configured to determine a debugging link of the controller to be debugged according to the identification of the controller to be debugged; and transmit the debugging data to the controller to be debugged through the debugging link.

[0104] In some examples, the first receiving module 1201 is further configured to determine a project identification of the controller to be debugged and an identification of a client side to which the controller to be debugged belongs, according to the identification of the controller to be debugged; and generate the debugging link based on the identification and the project identification of the controller to be debugged, and the identification of a target client side to which the controller to be debugged belongs.

[0105] In some examples, the remote debugging device 1200 further comprises an identification judging module 1204, configured to judge whether the first operating data or the second operating data carries the project identification and the identification of the client side; and adding the project identification and/or the identification into the first operating data or the second operating data in response to the determination that the project identification and/or the identification of the client side are/is not carried.

[0106] In some examples, the data transmitting mod-

ule 1203 is further configured to acquire a debugging type of the debugging instruction, acquire a method base required to be called for the debugging type according to the debugging type, and generate a data architecture, for debugging correspondingly, as the debugging data.

[0107] In some examples, the remote debugging device 1200 further comprises a collating module, configured to collate the first operating data or the second operating data to obtain key information; and to adjust the key information structurally to generate a target data architecture.

[0108] Figure 13 is a schematic diagram showing a remote debugging device according to at least one of examples of the present disclosure. As shown in Figure 13, the remote debugging device 1300 is installed at a client side, comprising a transmitting module 1301 and a receiving module 1302.

[0109] The transmitting module 1301 is configured to transmit first operating data of a controller.

[0110] The receiving module 1302 is configured to receive debugging data corresponding to a controller to be debugged, and debug the controller to be debugged according to the debugging data.

[0111] In some examples, the receiving module 1302 is further configured to transmit second operating data generated, by the controller to be debugged, during and/or after a debugging process.

[0112] In some examples, the operating data of the controller comprises resource usage state data and/or configuration information of the controller to be debugged.

[0113] In some examples, the receiving module 1302 is further configured to parse the debugging data to generate discrete debugging data; process and regroup the discrete debugging data to generate regrouped debugging data; acquire a debugging type of the regrouped debugging data, acquire an event library required to be called for the debugging type according to the debugging type, and generate second operating data of the controller to be debugged; and transmit the second operating data.

[0114] To achieve the above examples, the present disclosure further provides in examples an electronic device 1400. As shown in Figure 14, the electronic device 1400 comprises: a processor 1401; and a memory 1402, communicated to the processor, where the memory 1402 has stored therein executable instructions that, when executed by the processor 1401, achieve a remote debugging method according to the above examples.

[0115] To achieve the above examples, the present disclosure further provides in examples a non-transitory computer-readable storage medium having stored therein computer executable instructions that, when executed by a computer, achieves a remote debugging method according to the above examples.

[0116] To achieve the above examples, the present disclosure further provides in examples a computer program product, comprising a computer program that,

when executed by a processor, achieves a remote debugging method according to the above examples.

[0117] To achieve the above examples, the present disclosure further provides in examples a computer program, comprising a computer program code that, when executed by a computer, implements a remote debugging method according to the above examples.

[0118] It should be noted that the interpretation for a remote debugging method according to the above examples applies in a remote debugging device, an electronic device, a non-transitory computer-readable storage medium, a computer program product and a computer program according to the above examples, which will not be repeated herein.

[0119] In the specification, it should be understood that, the terms indicating orientation or position relationship such as "central", "longitudinal", "lateral", "width", "thickness", "above", "below", "front", "rear", "right", "left", "vertical", "horizontal", "top", "bottom", "inner", "outer", "clockwise", "counter-clockwise", "axial", "radial", "circumferential" should be construed to refer to the orientation or position relationship as then described or as shown in the drawings. These terms are merely for convenience and concision of description and do not alone indicate or imply that the device or element referred to must have a particular orientation or must be configured or operated in a particular orientation. Thus, it cannot be understood to limit the present disclosure.

[0120] In addition, terms such as "first" and "second" are used herein for purposes of description and are not intended to indicate or imply relative importance or significance or impliedly indicate quantity of the technical feature referred to. Thus, the feature defined with "first" and "second" may comprise one or more this feature. In the description of the present disclosure, "a plurality of" means two or more than two this features, unless specified otherwise.

[0121] Reference throughout this specification to "an embodiment," "some embodiments," "an example," "a specific example," or "some examples," means that a particular feature, structure, material, or characteristic described in connection with the embodiment or example is comprised in at least one embodiment or example of the present disclosure. Thus, the appearances of the phrases such as "in some embodiments," "in one embodiment", "in an embodiment", "in another example," "in an example," "in a specific example," or "in some examples," in various places throughout this specification are not necessarily referring to the same embodiment or example of the present disclosure. Furthermore, the particular features, structures, materials, or characteristics may be combined in any suitable manner in one or more embodiments or examples. Besides, any different embodiments and examples and any different characteristics of embodiments and examples may be combined by those skilled in the art without contradiction.

[0122] Although explanatory embodiments have been shown and described, it would be appreciated by those

skilled in the art that the above embodiments cannot be construed to limit the present disclosure, and changes, alternatives, and modifications can be made in the embodiments in the scope of the present disclosure.

5

Claims

1. A method for remote debugging, for a cloud server side, comprising:

10

receiving a first operating data of a controller; acquiring the first operating data of a controller to be debugged according to a debugging instruction, wherein the debugging instruction comprises an identification of the controller to be debugged; and generating a debugging data according to the first operating data, and transmitting the debugging data to the controller to be debugged.

15

20

2. The method according to claim 1, wherein subsequent to transmitting the debugging data to the controller to be debugged, the method further comprises:

25

receiving a second operating data generated, by the controller to be debugged, during and/or after a debugging process; and generating a debugging log of the controller to be debugged according to the second operating data.

30

3. The method according to claim 1 or 2, wherein the operating data of the controller comprises a resource usage state data of the controller to be debugged and/or a configuration information of the controller to be debugged.

35

4. The method according to any one of claims 1 to 3, wherein transmitting the debugging data to the controller to be debugged, comprises:

40

determining a debugging link of the controller to be debugged according to the identification of the controller to be debugged; and transmitting the debugging data to the controller to be debugged through the debugging link.

45

5. The method according to any one of claims 1 to 4, wherein subsequent to acquiring the first operating data of a controller to be debugged, the method further comprises:

50

determining a project identification of the controller to be debugged and an identification of a client side, according to the identification of the controller to be debugged; and

55

generating the debugging link based on the identification of the controller to be debugged, the project identification of the controller to be debugged, and the identification of the target client side.

6. The method according to any one of claims 2 to 5, wherein subsequent to acquiring the first operating data or the second operating data, the method further comprises:

judging whether the first operating data or the second operating data carries the project identification and the identification of the client side; and adding the project identification and/or the identification of the client side into the first operating data or the second operating data in response to the determination that the project identification and/or the identification of the client side are/is not carried.

7. The method according to any one of claims 1 to 6, wherein prior to transmitting the debugging data to the controller to be debugged, the method further comprises:

acquiring a debugging type of the debugging instruction, acquiring a method base required to be called for the debugging type according to the debugging type, and generating a data architecture corresponding for debugging as the debugging data.

8. The method according to any one of claims 2 to 7, wherein subsequent to acquiring the first operating data or the second operating data, the method further comprises:

collating the first operating data or the second operating data to obtain a key information; and adjusting the key information structurally to generate a target data architecture.

9. A method for remote debugging, for a client side, comprising:

transmitting a first operating data of a controller; and receiving a debugging data corresponding to a controller to be debugged, and debugging the controller to be debugged according to the debugging data.

10. The method according to claim 9, wherein subsequent to receiving a debugging data corresponding to a controller to be debugged, and debugging the controller to be debugged according to the debugging data, the method further comprises:

transmitting a second operating data generated, by

the controller to be debugged, during and/or after a debugging process.

11. The method according to claim 9 or 10, wherein the operating data of the controller comprises a resource usage state data and/or a configuration information of the controller to be debugged. 5

12. The method according to any one of claims 9 to 11, wherein subsequent to receiving a debugging data corresponding to a controller to be debugged, the method further comprises: 10

- parsing the debugging data to generate a discrete debugging data; 15
- processing and regrouping the discrete debugging data to generate a regrouped debugging data;
- acquiring a debugging type of the regrouped debugging data, acquiring an event library required to be called for the debugging type according to the debugging type, and generating a second operating data of the controller to be debugged; 20
- and
- transmitting the second operating data. 25

13. A remote debugging device, for a cloud server side, comprising:

- a first receiving module, configured to receive a first operating data of a controller; 30
- a data acquiring module, configured to acquire the first operating data of a controller to be debugged according to a debugging instruction, wherein the debugging instruction comprises an identification of the controller to be debugged; 35
- and
- a data transmitting module, configured to generate a debugging data according to the first operating data, and transmit the debugging data to the controller to be debugged. 40

14. A remote debugging device, for a client side, comprising: 45

- a transmitting module, configured to transmit a first operating data of a controller; and
- a receiving module, configured to receive a debugging data corresponding to a controller to be debugged, and debug the controller to be debugged according to the debugging data. 50

15. An electronic device, comprising:

- at least a processor; and 55
- a memory, communicated to said at least one processor,
- wherein the memory has stored therein an ex-

ecutable instruction that, when executed by said at least one processor, achieve a remote debugging method according to any one of claims 1 to 8 or 9 to 12.

16. A non-transitory computer-readable storage medium having stored therein a computer executable instruction that, when executed by a computer, achieves a remote debugging method according to any one of claims 1 to 8 or 9 to 12.

17. A computer program product, comprising a computer program that, when executed by a processor, achieves a remote debugging method according to any one of claims 1 to 8 or 9 to 12.

18. A computer program, comprising a computer program code that, when executed by a computer, implements a remote debugging method according to any one of claims 1 to 8 or 9 to 12.

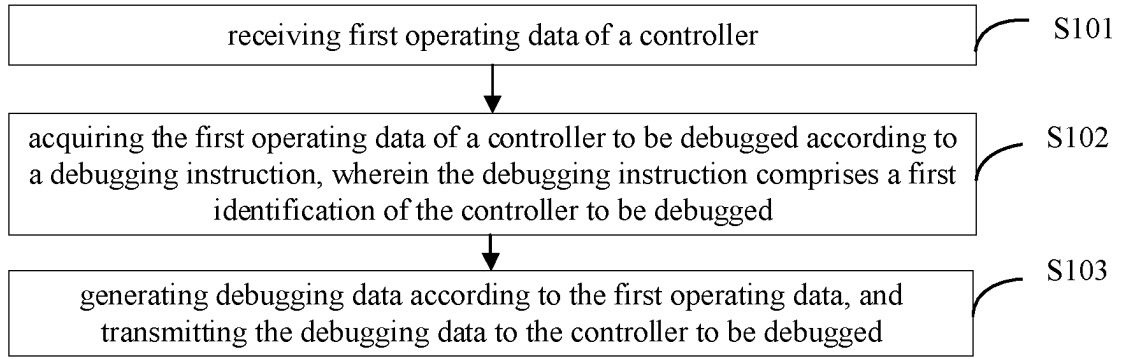


Figure 1

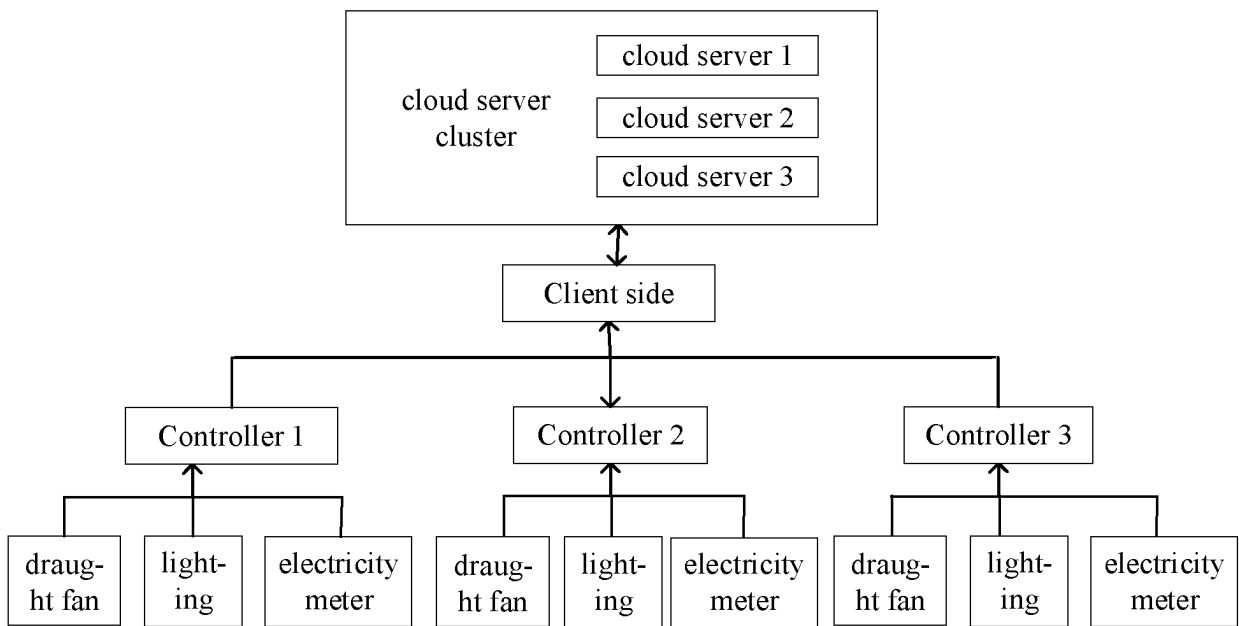


Figure 2

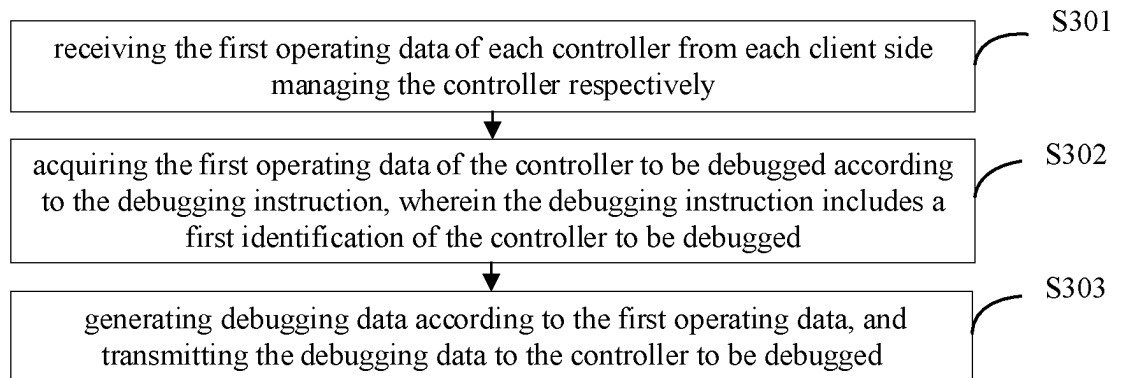


Figure 3

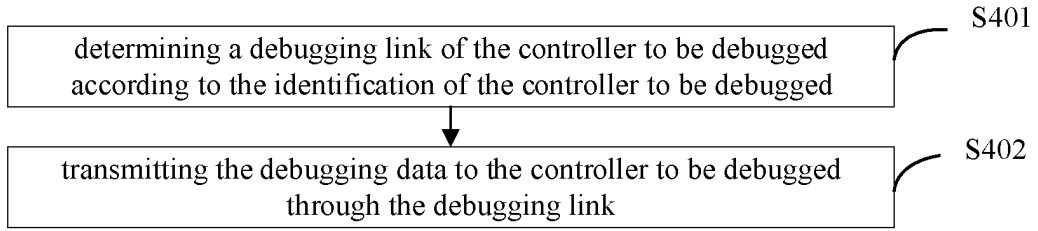


Figure 4

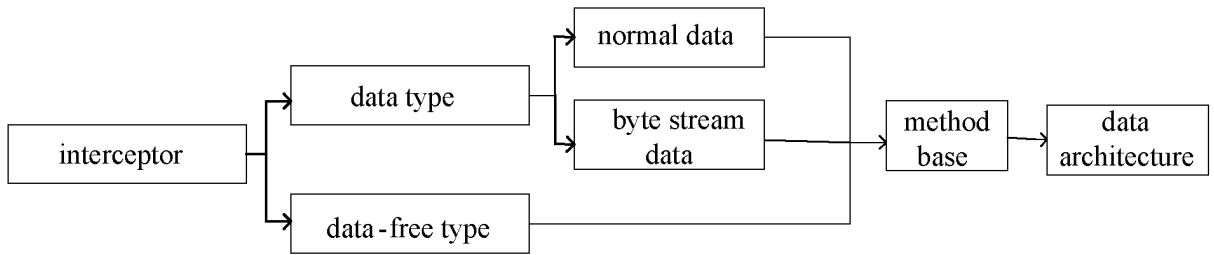


Figure 5

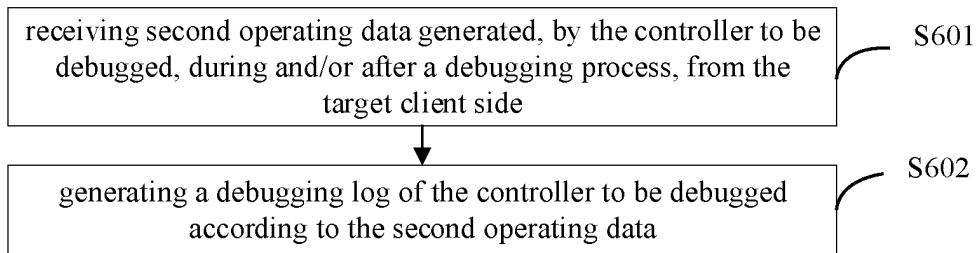


Figure 6

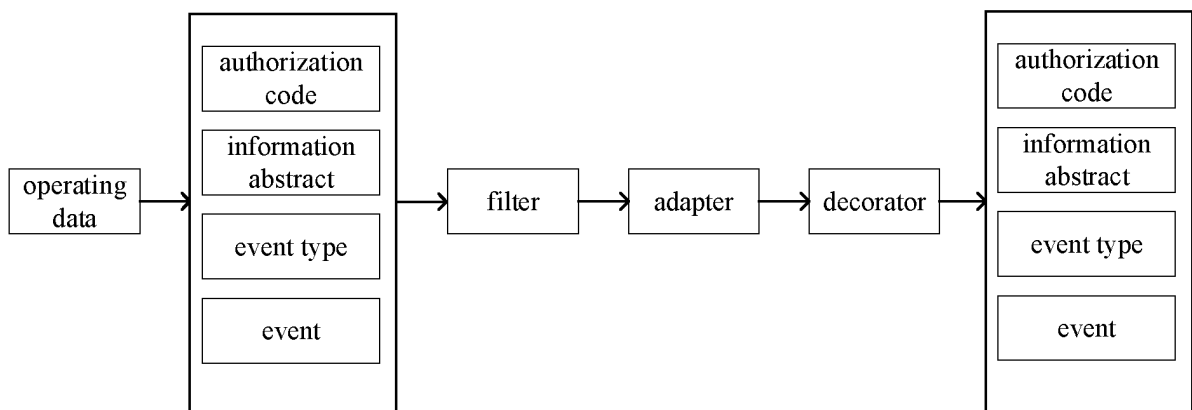


Figure 7

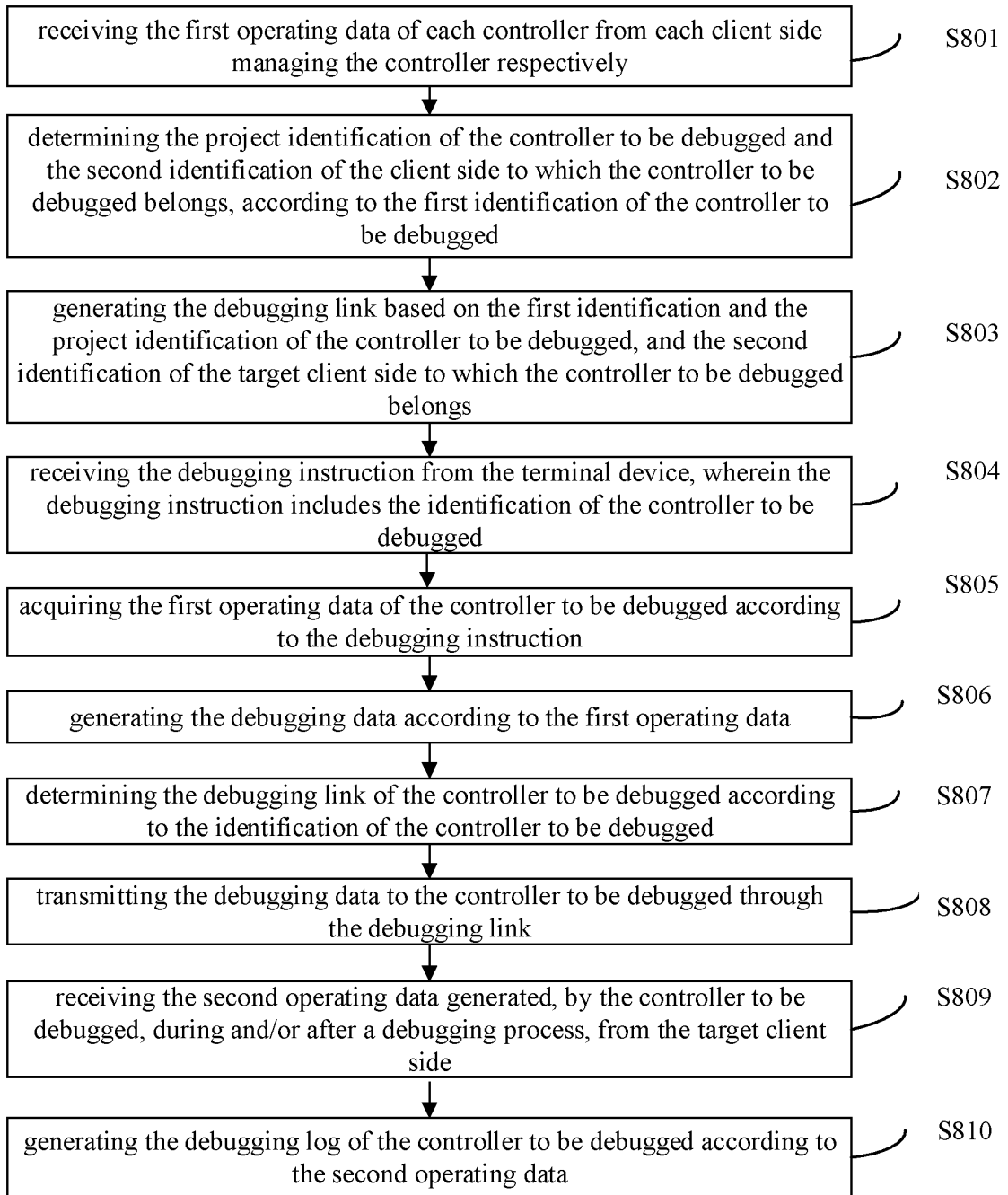


Figure 8

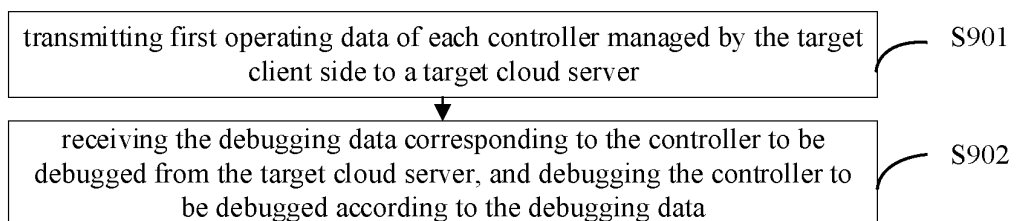


Figure 9

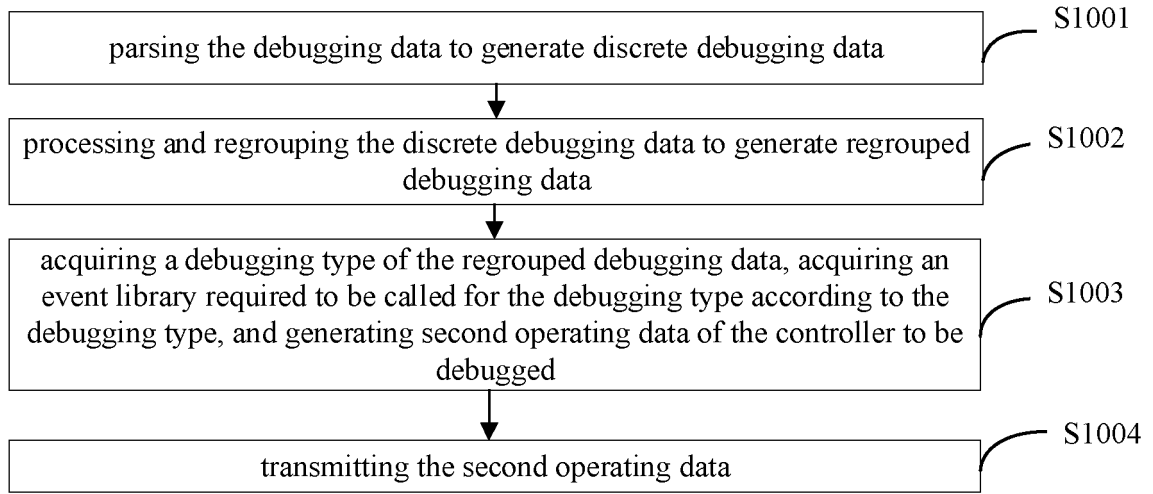


Figure 10

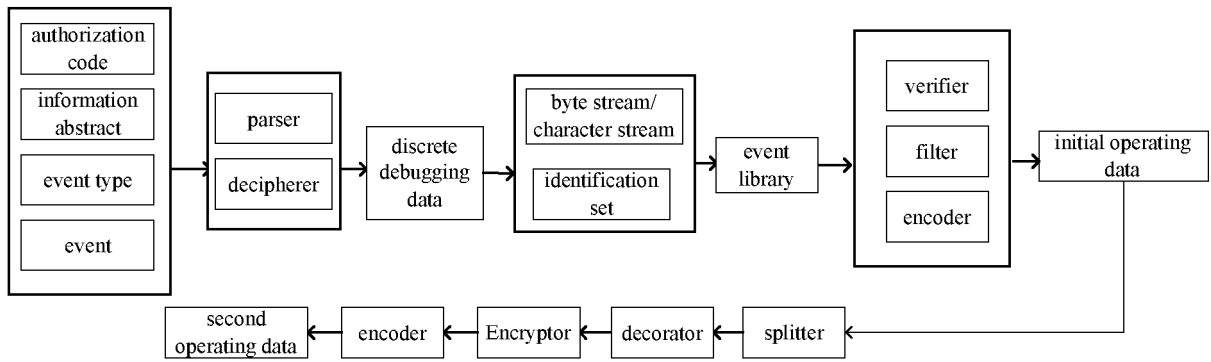


Figure 11

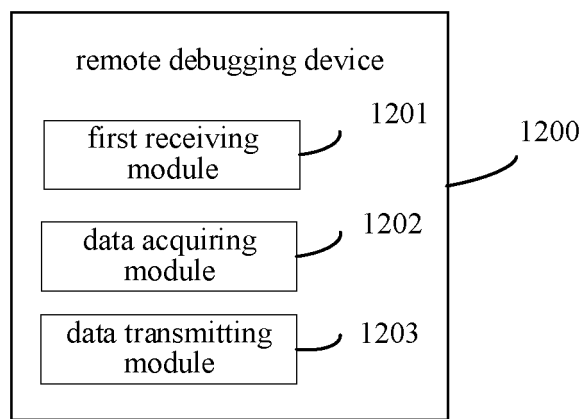


Figure 12

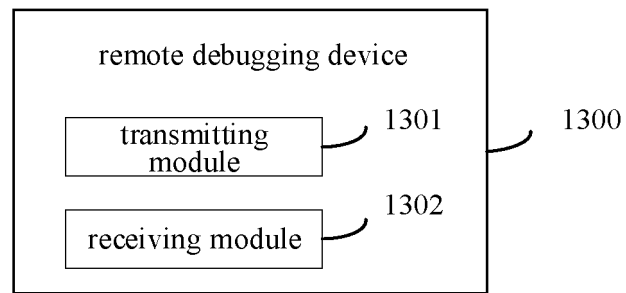


Figure 13

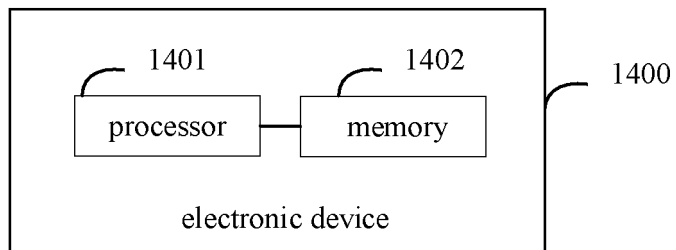


Figure 14

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2022/117083

5

A. CLASSIFICATION OF SUBJECT MATTER

G05B 23/02(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

10

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G05B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

15

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, CNKI, WPI, EPODOC: 远程 or 远距离, 调试, 设备 or 现场 or 控制器 or 本地, 名称 or 标识 or 命名 or 识别 or 地址, 服务器 or 云, 日志 or 记录, debug+, remote, sever? or cloud+, name? or ID or IP or address+, log or record+

20

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 103472790 A (SUZHOU INOVANCE TECHNOLOGY CO., LTD. et al.) 25 December 2013 (2013-12-25) description, paragraphs [0043]-[0071], and figures 1-6	1-18
X	CN 103466401 A (SUZHOU INOVANCE TECHNOLOGY CO., LTD. et al.) 25 December 2013 (2013-12-25) description, paragraphs [0026]-[0041], and figures 1-3	1-18
A	CN 1703870 A (CISCO TECHNOLOGY INC.) 30 November 2005 (2005-11-30) entire document	1-18
A	CN 110413329 A (BEIJING ANYUN CENTURY TECHNOLOGY CO., LTD.) 05 November 2019 (2019-11-05) entire document	1-18
A	CN 112235417 A (BEIJING BAIDU NETCOM SCIENCE AND TECHNOLOGY CO., LTD.) 15 January 2021 (2021-01-15) entire document	1-18
A	CN 107807545 A (BEIJING JINGDONG SHANGKE INFORMATION TECHNOLOGY CO., LTD. et al.) 16 March 2018 (2018-03-16) entire document	1-18

35

 Further documents are listed in the continuation of Box C.
 See patent family annex.

40

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

45

Date of the actual completion of the international search

09 November 2022

Date of mailing of the international search report

29 November 2022

50

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/
CN)
No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing
100088, China

Authorized officer

55

Facsimile No. (86-10)62019451

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2022/117083

5
10
15
20
25
30
35
40
45
50
55

C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 214480706 U (CHUANGXINGONGFANG TECHNOLOGY (SHENZHEN) CO., LTD.) 22 October 2021 (2021-10-22) entire document	1-18
A	US 2010083356 A1 (STECKLEY, A. et al.) 01 April 2010 (2010-04-01) entire document	1-18
A	EP 3404898 A1 (MEAS FRANCE) 21 November 2018 (2018-11-21) entire document	1-18

Form PCT/ISA/210 (second sheet) (January 2015)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2022/117083

5

10

15

20

25

30

35

40

45

50

55

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 103472790 A	25 December 2013	None	
CN 103466401 A	25 December 2013	CN 103466401 B	25 January 2017
CN 1703870 A	30 November 2005	AU 2003279138 A1	04 May 2004
		DE 60323252 D1	09 October 2008
		AT 406729 T	15 September 2008
		WO 2004034638 A2	22 April 2004
		CA 2499336 A1	22 April 2004
		US 2004073658 A1	15 April 2004
		EP 1550263 A2	06 July 2005
		EP 1550263 B1	27 August 2008
		CN 100446478 C	24 December 2008
		AU 2003279138 B2	05 March 2009
CN 110413329 A	05 November 2019	None	
CN 112235417 A	15 January 2021	CN 112235417 B	10 June 2022
		US 2021263817 A1	26 August 2021
CN 107807545 A	16 March 2018	CN 107807545 B	30 March 2021
CN 214480706 U	22 October 2021	None	
US 2010083356 A1	01 April 2010	WO 2010037126 A2	01 April 2010
		US 9210220 B2	08 December 2015
EP 3404898 A1	21 November 2018	EP 3404898 B1	23 September 2020

Form PCT/ISA/210 (patent family annex) (January 2015)

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- CN 202111283676 [0001]